



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Table of Contents

THIS POLICY APPLIES TO ALL DES ADMINISTRATIVE ENTITIES, COUNCILS, DIVISIONS, ADMINISTRATIONS, AND PROGRAMS.**ERROR! BOOKMARK NOT DEFINED.**

1. PURPOSE	5
2. BACKGROUND	5
3. UNDERSTANDING DISABILITY ISSUES WHEN DESIGNING WEB SITES	5
4. VISUAL.....	6
5. HEARING	6
6. MOBILITY.....	7
7. COGNITIVE AND LEARNING DISABILITIES.....	7
8. ASSISTIVE TECHNOLOGIES	7
10. GETTING INTO JAWS.....	7
11. ENTERING FORMS MODE	8
12. NAVIGATE BETWEEN FRAMES.....	8
13. NAVIGATE BETWEEN A LIST OF HYPERLINKS	8
14. NAVIGATE BETWEEN A LIST OF FORM FIELDS	8
15. USING A DROP DOWN BOX	8
16. JAWS NAVIGATION KEYSTROKES.....	9
17. HOW TO USE THIS DOCUMENT	9
18. WEB FILE FORMATS.....	10
(a) <i>Adobe Portable Document (PDF) Files</i>	<i>10</i>
(a) <i>Rationale.....</i>	<i>10</i>
(b) <i>Techniques.....</i>	<i>10</i>
(c) <i>Testing</i>	<i>11</i>



Department of Economic Security

Version 1.0

Subject: Web Accessibility Guidelines – For Developers

19. FORMS.....	11
(b) Rationale.....	11
(a) Techniques.....	11
(b) Example:.....	11
(c) Testing.....	12
20. TABLES.....	12
(a) Rationale.....	12
(b) Techniques.....	12
(c) Example:.....	12
(d) Example:.....	14
Meals.....	14
Hotels.....	14
Transport.....	14
Totals.....	15
21. USE THE SUMMARY ATTRIBUTE TO DESCRIBE THE INTENT OF THE DESIGN FOR A NON-TABULAR TABLE.....	16
(a) Example:.....	16
(b) Testing.....	16
(c) References.....	16
22. FRAMES.....	17
(a) Rationale.....	17
(b) Techniques.....	17
(c) Example:.....	17
(d) Testing.....	17
CHECK THAT EACH FRAME PAGE HAS A MEANINGFUL TITLE IN THE <TITLE> TAG.....	17
23. REFERENCES:	17
(a) Frame title.....	17
24. SCRIPTS, APPLETS AND PLUG-INS	17
(a) Rationale.....	18
(b) Techniques.....	18
(c) Scripts.....	18
(d) Applets and Plug-ins.....	18
(e) Example:.....	18
(f) Example:.....	19
(g) Java Accessibility.....	19
(h) Testing.....	19
(i) References.....	20
25. NON-TEXT ELEMENTS.....	20
(a) Rationale.....	20



Department of Economic Security

Version 1.0

Subject: Web Accessibility Guidelines – For Developers

(b)	Techniques.....	20
(c)	Example:.....	20
(d)	Example:.....	20
(e)	Example:.....	21
(f)	Example:.....	21
(g)	Testing	21
(h)	References.....	21
26.	IMAGE MAPS.....	21
(a)	Rationale.....	21
(b)	Techniques.....	22
(c)	Example:.....	22
(d)	Example:.....	22
(e)	Testing	22
(f)	Reference	23
27.	MULTIMEDIA	23
(a)	Rationale.....	23
(b)	Techniques.....	23
(c)	Testing	23
(d)	References.....	24
28.	COLOR	24
(a)	Rationale.....	24
(b)	Techniques.....	24
(c)	Testing	24
(d)	References.....	24
29.	NAVIGATION AND DESIGN.....	24
(a)	Rationale.....	25
(b)	Techniques.....	25
(c)	Example:.....	26
(d)	Example:.....	26
(e)	Testing	26
(f)	References.....	26
30.	GRAPHS AND CHARTS.....	26
(a)	Surrounding Description Text	27
(b)	Long Description	27
(c)	D-Link.....	28
31.	STYLE SHEETS.....	29
(a)	Rationale.....	29
(b)	Techniques.....	29
(c)	Example:.....	29
(d)	Example:.....	30
(e)	Testing	30



Department of Economic Security

Version 1.0

Subject: Web Accessibility Guidelines – For Developers

(f)	References.....	31
32.	FLASH MOVIES & ANIMATION.....	31
(a)	Understanding How Assistive Technologies Handle Animation	31
33.	PROVIDING TEXT EQUIVALENTS FOR ENTIRE MOVIES.....	32
34.	MAKING CHILD OBJECTS INACCESSIBLE	32
35.	AVOID CONSTANT MOTION	32
36.	ACCESSIBILITY AND MACROMEDIA FLASH PLAYER	33
37.	ENHANCED ACCESSIBILITY	33
(b)	System Requirements.....	33
(c)	Accessing Macromedia Flash Content	34
38.	APPENDIX.....	35
(a)	Validation and Testing.....	35
39.	W3C'S WEB ACCESSIBILITY TUTORIAL	35
40.	WEB ACCESSIBILITY CHECKLIST	36
41.	CONCLUSION.....	37



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

1. Purpose

These guidelines provide specific direction for the universal accessibility of Web-related pages, Web applications, static web pages, PDF documents for displaying data and information for business partners and customers, regardless of their abilities or disabilities. These guidelines will help to ensure that Web-applications and static pages are compatible with the JAWS screen reading software and other assistive technologies.

The DES considered information from various sources to develop these guidelines, including the following:

- Architectural and Transportation Barriers Compliance Board (Access Board), Electronic and Information Technology Accessibility Standard, §1194.22 (Web-based Intranet and Internet Information and Applications) published in the *Federal Register* (65 FR 80500, December 21, 2000).
- General Services Administration.
- World Wide Web Consortium (W3C).
- Department of Education.
- Other government agencies.
- Private industry.
- JAWS technical support team.
- Bobby software for evaluating website accessibility compliance.

2. Background

The Workforce Investment Act (WIA) of 1998 included Section 508 of the Rehabilitation Act Amendments of 1998. This requires Federal departments, state governments and agencies to ensure that their electronic and information technology (EIT) is accessible to persons with disabilities, including employees and the public. The WIA's reference to electronic and information technology is interpreted to include Web sites, facsimile machines, copiers, software, computers, and other similar technologies.

Section 508 requires federal agencies to procure, develop, maintain, and use EIT that is accessible to all persons, unless meeting the requirements will cause an undue burden on the department or agency. *Undue burden* generally means a significant difficulty or expense. The agency claiming an undue burden must do the following:

- Provide documentation explaining why compliance creates an undue burden.
- Provide an alternate format or method for accessing the information or data.

3. Understanding disability issues when designing web sites

When you design or modify Web sites to allow access to people with disabilities, you make the Web accessible. New Web sites and applications, however, are introducing new problems and barriers. There are complex graphics and multimedia applications that assistive technology simply has not solved. One solution to these new problems is to put accessibility in the hands of the Web developer and content author. Creating a Web site that is accessible by people with disabilities is relatively easy as long as the Web developer and author follow some basic guidelines. Assistive technology is a piece of equipment or a software product that is used to increase, maintain, or assist the functional capabilities of individuals with



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

disabilities. In short, it can be any device or technique that assists people in removing or reducing barriers and enhancing their daily activities. Assistive technologies include magnifiers, screen readers, closed captioning, keyboard enhancements, and highlighting software. Assistive technology uses the coding and content of your Web site and makes it accessible. This course is not designed to make you an "assistive technology" developer, but does provide steps and guidelines to follow that address the accessibility needs of users.

Meeting the standards of an accessible Web site first requires an awareness of the special needs of users who have disabilities. The four main categories of disabilities are visual, hearing, mobility, and cognitive and learning disabilities. Each person with a disability might encounter one or more barriers that can be eliminated or minimized by the Web developer, the browser, the assistive technology, or the underlying operating system software and hardware platform.

4. Visual

People with visual disabilities are individuals who are blind, have low vision, or have color blindness. People who are blind need text equivalents for the images used on the Web page, because they and their assistive screen reader technology cannot obtain the information from the image. A person who has a visual disability will not find the mouse useful because it requires hand and eye coordination. Instead, this person must navigate the Web page using only the keyboard. For example, the Tab key is used to move the focus to the item that needs to be selected. A screen reader then announces the item so the user knows where the focus is on the page. The user then presses the Enter key instead of "clicking" the mouse button. Those who have low vision need the assistance of a hardware or software magnifier to enlarge the text beyond simple font enlargement. People who are colorblind or who have low vision benefit from good contrasting colors. When information is presented by color alone, a person who is colorblind misses that information. Similarly, if information is presented using any attribute by itself (for example, contrast, depth, size, location, or font), a user who has low vision might not detect the difference.

Magnification might reformat the location, change the contrast, or distort the size and fonts of the text and objects on the Web page. It is best to use multiple attributes. For example, if both color and a fill pattern are used on different bars on a graph, they can be viewed in either color or black and white. Instead of using size attributes on the font element to denote a heading, the heading element should be used to correctly mark up a heading so that assistive technology can identify headings.

5. Hearing

People who are deaf or hard of hearing require visual representations of auditory information that the Web site provides. Solutions for these disabilities include closed captioning, blinking error messages, and transcripts of the spoken audio. The primary concern is to ensure that audio output information is provided in a redundant equivalent visual form.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

6. Mobility

People with mobility disabilities have physical impairments that substantially limit movement and fine motor controls, such as lifting, walking, and typing. Mobility impaired individuals experience difficulties in using the computer's input devices and in handling storage media. Solutions for persons with mobility disabilities include switches, latches, and controls that are easy to manipulate, and diskettes and media that are easy to insert and remove. Additional solutions include alternate input capabilities, such as voice input or the ability to enter information at the user's own pace. For example, sequences of keystrokes can be typed, one at a time, rather than simultaneously as in Ctrl+Alt+Del. Many of these needs are supported by assistive technology, operating systems, and hardware platforms. Furthermore, making the Web site accessible will make it more compatible with voice input and control technologies.

7. Cognitive and Learning Disabilities

People with cognitive or learning disabilities, such as dyslexia and short-term memory deficit, need more general solutions, which include providing a consistent design and using simplified language. For example, by using a template, a Web developer can reuse the same layout and design for each page, so a person with a cognitive disability can more easily navigate through a Web site. People with cognitive or learning disabilities can also benefit from redundant input, such as providing both an audio file and a transcript of a video. By simultaneously viewing the text and hearing it read aloud, they could take advantage of both auditory and visual skills to comprehend the material better.

8. Assistive Technologies

In the broadest sense, an assistive technology is any electronic or information technology that enables a person to do something they otherwise could not have done. A variety of assistive technologies currently provide access to the computer system.

Examples of assistive technologies include the following:

- A screen reader, which is a speech synthesizer that converts text to speech.
- A screen magnifier, which enlarges the screen display.
- Keyboard alternatives such as split or natural keyboards.
- Shortcut keys for all mouse actions.

The scope for this document on assistive technologies is restricted only to the JAWS screen reader software.

9. Screen Reader Software

[JAWS Standard for Windows](http://www.hj.com/)

<http://www.hj.com/> or <http://www.freedomscientific.com/>

10. Getting into JAWS

JAWS for Windows provides speech technology that works with the Windows operating system to access popular software applications and the Internet. This research will help us find out how the JAWS software supports HTML frames in a web page. How does it read through data in the



Department of Economic Security

Version 1.0

Subject: Web Accessibility Guidelines – For Developers

different frames of the web page? And any possible ways to find out if a web page is compliant to W3C & WAI standards.

JAWS is in compliance with the WAI (Web accessibility initiative, <http://www.w3.org/WAI/>) at the W3 consortium.

JAWS had support for HTML frames right from the version 3.2 where you can hit Control + tab keys on your keyboard to navigate between the frames and the software would read out all the information on the selected frame. When you hit control + tab keys it JAWS will announce the name of the frame you entered in. JAWS announces the beginning and end of frames, reading the frame title if one is available. If no frame title is available, JAWS reads the name of the selected frame.

11. Entering FORMS Mode

Initially when a web page is loaded it is not set to default to forms mode. Forms mode enables a user to tab through every form field & hyperlink on the web page. To switch to forms mode, hit one TAB on the web page, which will take you to the first control on the web page. A control can be a form field or a hyperlink or any object that can get focus from the user. Once you have hit the TAB key hit the ENTER key. JAWS will announce that it has entered the FORMS mode. Now the user can hit TAB and navigate between fields and make data entry.

12. Navigate between frames

Press INSERT + F9

Scroll through the frames list using UP & DOWN arrow keys. JAWS will read the name of the frames. It is important to assign a sensible name to every frame in the page so that the user can understand it.

Press the ENTER key when you hear the frame name you want to navigate to.

13. Navigate between a list of Hyperlinks

Press INSERT + F7

Scroll through the hyperlink list using UP & DOWN arrow keys. JAWS will read the name of the hyperlinks. You can describe the functionality of every hyperlink with the help of the ALT attribute in the tag

Press the ENTER key when you hear the hyperlink name you want to navigate to.

14. Navigate between a list of Form Fields

Press INSERT + F5

Scroll through the form field list using UP & DOWN arrow keys. JAWS will read the name of the field. You can describe the functionality of every hyperlink with the help of the ALT attribute in the <input type=text id="txt_name" alt="First Name" > tag

Press the ENTER key when you hear the hyperlink name you want to navigate to.

15. Using a drop down box



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

After the focus is set to a drop down by either TABing or by using the forms list pressing INSERT F5, hit the ALT + DOWN arrow key. You can select different items in the drop down using UP & DOWN arrow keys.

As in the previous versions we have seen that JAWS was trying to pronounce the State codes like AL. But in the newer versions when you are on a particular ITEM in the drop down, hit the INSERT + NUM PAD number 5 twice quickly. This will spell the selected ITEM and not pronounce.

Press the ENTER key to select the item from the drop down.

16. JAWS Navigation Keystrokes

There are different keystrokes for the Desktop PC keyboard and the Laptop keyboard and the brail keyboards.

Other Software Add-In/Plug-In's (Acrobat PDF support). You can also download the latest add-ins for other application support like Acrobat PDF and MSN Messenger, Winzip, MSNBC News alerts, Roxie & Easy CD creator. You can download these additional plug-ins from <http://www.jfwlite.com/programs.html>.

17. How to Use This Document

This document is designed as a technical reference document, which uses navigation links, in the form of a "Table of Contents", to quickly link to a specific Web accessibility guideline within this document.

Each guideline includes a general description, followed by these four sections:

Rationale: In this section, we describe how the issue impacts certain disabilities or assistive technologies

Techniques: In this section, we describe accepted design practices or alternate sources for providing the same information.

Testing: In this section, we provide methods for evaluating accessibility solutions. The testing methods will involve an assistive technology tool such as a screen reader. The screen reader will aid in determining whether the information can be read, interpreted correctly, and is in compliance with the Access Board's guidelines.

References: In this section, we provide Web-based links to additional information related to a specific guideline. We are providing these references to developers to help them understand web design techniques.

Some guidelines reference hypertext markup language (HTML) code and include the following formatting conventions:

<TAG> References a specific HTML tag.

Attribute References a specific HTML attribute.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

18. Web File Formats

Adobe Portable Document (PDF) Files

An alternative format should be provided either in addition to an Adobe Portable Document Format (PDF) file or in place of the PDF file.

(a) Rationale

PDF files are single graphic images that are not accessible by screen readers. Although there have been attempts to use a plug-in to make PDF files accessible to screen readers, these attempts have produced less than desirable results. Various vendors have also attempted to provide tools that convert PDF files and make them accessible to screen readers. Conversion tools work best with simple text documents that contain little formatting. They do not provide a total conversion for documents that contain tables, images, charts, bullets, etc.

(b) Techniques

First, review the PDF files and remove any PDF files that are obsolete or no longer provide valuable information.

For those PDFs that remain, provide an alternate format that will be used in addition to, or in place of the PDF file when compliance cannot be accomplished in any other way. Alternate formats are intended to have equivalent content and functionality to the primary Web source and must be updated concurrently with the primary Web source.

There are two options for generating an alternate format. These options depend on the availability of the original file that was used to generate the PDF file.

In the first option, the original source file that generated the PDF is available. This file is used to create the alternate format. For internal or external Web sites, that alternate format may be an HTML, plain text, or rich text format (RTF) file. For internal sites only, the alternate format may also be one of the native formats used by the standard office products (e.g., Word, Excel, and PowerPoint).

In the second option, the original source file used to generate the PDF is not available. A conversion tool is needed to produce either an HTML or plain text file. Currently conversion software is not 100 percent accurate. For example, not all tables and columns convert properly or in the correct order. Images, charts, and graphics will not convert. Detailed editing and validation must be done to ensure that the content of the converted document is the same as the source file.

PDF forms available for download must have an associated text description that describes the form number, title, and purpose.

In considering the options suggested above, keep in mind the extent of the conversion and review efforts; the size of the alternate file format and the time that it will take to load



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

or download the file; the availability of a source file; and the relevancy of existing PDF files. The conversion of PDF files requires extensive review, comparison, and validation.

There are other solutions to providing alternate formats that do not involve files. Examples of these include Braille printouts, direct telephone support, and one-on-one explanations.

(c) Testing

Use a screen reader to determine if the information is easily comprehensible to the visually impaired. Verify that the content of the alternate format or the converted file is equal to the content and functionality of the primary PDF file and is updated concurrently with the primary Web source. Ensure that no information has been lost and that the meaning has remained the same.

19. Forms

When electronic forms are designed to be completed online, the form must allow access via assistive technologies to information, field elements, and functionality (e.g., associated controls required for completion and submission of the form, including directions and cues).

Rationale

Screen readers may have difficulty navigating through interactive forms without some modification to the form.

Forms may require a timed response (e.g., security functions). Assistive technologies may prohibit users from completing that section of the form within the required time frame.

(a) Techniques

Construct electronic forms (for completion online) so that assistive technologies such as screen readers can relay the relevant field descriptors when tabbing from field to field. Use logical tabbing order consistently throughout the form.

Associate labels with all form elements. The label should appear either immediately to the left or immediately above the field. Additionally, all form controls (e.g., checkboxes and radio buttons), must have text descriptions. Some form controls automatically have text descriptions associated with them such as submit buttons. Image buttons also need descriptive text.

(b) Example:

Name:

Address:



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

```
<form method="post" ACTION="/cgi-bin/cgi.pl">
<p>
<LABEL for = "Name">Name:</LABEL>
<input tabindex="1" type="text" name="Name" size=30 maxlength=75><br>
<LABEL for = "Address"> Address:</LABEL>
<input tabindex = "2" type = "text" name = "Address" size =50  maxlength = 75> <p/> <p>
<input tabindex = "3" type = "submit" name = "Submit Button">
<input tabindex = "4" type = "reset" name = "Reset Button"> </p> </form>
```

When a timed response is required, the user must be alerted and given sufficient time to indicate that they need additional time to complete the form.

(c) Testing

Test all forms for accessibility using a screen reader. Verify that labels, text descriptions, tabbing order, etc., read correctly.

20. Tables

Tables must be constructed so that they can be easily interpreted by all users and communicate the intent of the author. Identify row and column headers for data tables. Associate data cells with their headers for all tables that have two or more logical levels of row or column headers. Non-tabular tables may be used for layout if the table uses appropriate HTML to summarize the intent of the design. Make sure that tables make sense when read from left to right, and from top to bottom.

(a) Rationale

Screen readers have difficulty interpreting data if the tables are not designed properly. A sighted person can scan down a column and across a row of a Postal Service rate table. This can be a daunting task for a visually impaired person to listen to this same information with a screen reader.

(b) Techniques

Use the <TABLE> tag primarily for the display of tabular data. Where appropriate, use the <CAPTION> tag and/or the *summary* attribute for tabular data. Also, use the *summary* attribute for non-tabular data. The content of the <CAPTION> tag is displayed on the screen. The contents of the *summary* attribute are read by the screen reader, but not displayed on the screen.

When a table has headers, use the table header <TH> tag and the *id* attribute to assign the header and identifier. Use the *headers* attribute in the <TD> tag to identify the column that the cell is located in. The following example demonstrates the use of the <CAPTION> and the <TH> tags as well as the *header*, *id*, and *summary* attributes.

(c) Example:



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Rate Table for Priority Mail Global Guaranteed

Weight not over (lbs.)	Rate Groups 1 and 2	Rate Groups 3 and 7	Rate Groups 4 and 6	Rate Group 5	Rate Group 8
0.5	\$20.00	\$24.00	\$29.00	\$40.00	\$60.00
1	\$31.00	\$34.00	\$41.00	\$51.00	\$72.00
2	\$31.00	\$34.00	\$41.00	\$65.00	\$87.00
3	\$42.00	\$53.00	\$60.00	\$80.00	\$102.00

<TABLE WIDTH = "66%" BORDER = "1" SUMMARY =This table summarizes the rates for Priority Mail Global Guaranteed postage rates.">

<CAPTION> Rate Table for Priority Mail Global Guaranteed</CAPTION>

<TR>

<TH ID= "HEADER1"> Weight not over (lbs)</TH>

<TH ID= "HEADER2"> Rate Groups 1 and 2</TH>

<TH ID= "HEADER3"> Rate Groups 3 and 7</TH>

<TH ID= "HEADER4"> Rate Groups 4 and 6</TH>

<TH ID= "HEADER5"> Rate Group 5</TH>

<TH ID= "HEADER6"> Rate Group 8</TH>

</TR>

<TR>

<TD ID= "ROW1"><CENTER>0.5</CENTER></TD>

<TD HEADERS= "ROW1 HEADER2"><CENTER> \$20.00</CENTER></TD>

<TD HEADERS= "ROW1 HEADER3"><CENTER> \$24.00</CENTER></TD>

<TD HEADERS= "ROW1 HEADER4"><CENTER> \$29.00</CENTER></TD>

<TD HEADERS= "ROW1 HEADER5"><CENTER> \$40.00</CENTER></TD>

<TD HEADERS= "ROW1 HEADER6"><CENTER> \$60.00</CENTER></TD>

</TR>

<TR>

<TD ID= "ROW2"><CENTER>1</CENTER></TD>

<TD HEADERS= "ROW2 HEADER2"><CENTER> \$31.00</CENTER></TD>

<TD HEADERS= "ROW2 HEADER3"><CENTER> \$34.00</CENTER></TD>

<TD HEADERS= "ROW2 HEADER4"><CENTER> \$41.00</CENTER></TD>

<TD HEADERS= "ROW2 HEADER5"><CENTER> \$51.00</CENTER></TD>

<TD HEADERS= "ROW2 HEADER6"><CENTER> \$72.00</CENTER></TD>

</TR>

<TR>

<TD ID= "ROW3"><CENTER>2</CENTER></TD>

<TD HEADERS= "ROW3 HEADER2"><CENTER> \$31.00</CENTER></TD>

<TD HEADERS= "ROW3 HEADER3"><CENTER> \$34.00</CENTER></TD>

<TD HEADERS= "ROW3 HEADER4"><CENTER> \$41.00</CENTER></TD>

<TD HEADERS= "ROW2 HEADER5"><CENTER> \$65.00</CENTER></TD>

<TD HEADERS= "ROW3 HEADER6"><CENTER> \$87.00</CENTER></TD>

</TR>



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

```

<TR>
  <TD ID= "ROW4"><CENTER>3</CENTER></TD>
<TD HEADERS= "ROW4 HEADER2"><CENTER> $42.00</CENTER></TD>
<TD HEADERS= "ROW4 HEADER3"><CENTER> $53.00</CENTER></TD>
<TD HEADERS= "ROW4 HEADER4"><CENTER> $60.00</CENTER></TD>
<TD HEADERS= "ROW4 HEADER5"><CENTER> $80.00</CENTER></TD>
<TD HEADERS= "ROW4 HEADER6"><CENTER> $102.00</CENTER></TD>
</TR>
</TABLE>

```

For data tables that have two or more logical levels, the following example shows how to create categories using the *axis* attribute.

(d) Example:

The image below shows how a browser might display this multi-dimensional table. The table lists travel expenses at two locations, San Jose and Seattle, by date, and category (meals, hotels and transport).

The caption centered above the table reads "Travel Expense Report".

The *axis* attributes or categories of the table are:

City.
Travel dates.
Meal expenses.
Hotel expenses.
Transport expenses.

The first row of the table contains these headers: "Meals", "Hotels", "Transport", and "subtotals". The first row group shows the expenses in San Jose for those categories on August 25 and 26, 1997. Below that row group, the subtotals of all expenses in San Jose are listed.

The second row group presents similar data for expenses incurred in Seattle on August 27 and 28, 1997. That group is followed by a row of subtotals as well.

The final row of the table presents totaled expenses (for all dates and cities) for "Meals", "Hotels", and "Transport".

Borders separate all cells, header or data.

Travel Expense Report

	Meals	Hotels	Transport	subtotals
San Jose				
25-Aug-97	37.74	112.00	45.00	
26-Aug-97	27.28	112.00	45.00	
379.02	65.02	224.00	90.00	379.02



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Seattle				
27-Aug-97	96.25	109.00	36.00	
28-Aug-97	35.00	109.00	36.00	
subtotals	131.25	218.00	72.00	421.25
Totals	196.27	442.00	162.00	800.27

<TABLE border ="1" summary = "This table summarizes travel expenses incurred during August trips to San Jose and Seattle">

<CAPTION>

Travel Expense Report

</CAPTION>

<TR>

<TH></TH>

<TH id = "a2" axis = "expenses">Meals</TH>

<TH id = "a3" axis = "expenses">Hotels</TH>

<TH id = "a4" axis = "expenses">Transport</TH>

<TD>subtotals</TD>

</TR>

<TR>

<TH id = "a6" axis = "location">San Jose</TH>

<TH></TH>

<TH></TH>

<TH></TH>

<TD></TD>

(a) </TR>

<TR>

<TD id="a7" axis="date">25-Aug-97</TD>

<TD headers="a6 a7 a2">37.74</TD>

<TD headers = "a6 a7 a3">112.00</TD>

<TD headers = "a6 a7 a4">45.00</TD>

<TD></TD>

</TR>

<TR>

<TD id="a8" axis="date">26-Aug-97</TD>

<TD headers = "a6 a8 a2">27.28</TD>

<TD headers = "a6 a8 a3">112.00</TD>

<TD headers = "a6 a8 a4">45.00</TD>

<TD ></TD>

</TR>

<TR>

<TD>subtotals</TD>

<TD headers = "a6 a8 a2"> 65.02</TD>

<TD>224.00</TD>

<TD>90.00</TD>

<TD>379.02</TD>

</TR>

<TR>

<TH id = "a10" axis = "location">Seattle</TH>

<TH></TH>

<TH></TH>



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

```
<TH></TH>
<TD></TD>
</TR>
<TD id="a11" axis="date">27-Aug-97</TD>
<TD headers="a10 a11 a2">96.25</TD>
<TD headers="a10 a11 a3">109.00</TD>
<TD headers="a10 a11 a4">36.00</TD>
<TD></TD>
</TR>
<TR>
  TD id="a12" axis="date">28-Aug-97</TD>
  <TD headers="a10 a12 a2">35.00</TD>
  <TD headers="a10 a12 a3">109.00</TD>
  <TD headers="a10 a12 a4">36.00</TD>
  <TD></TD>
</TR>
<TR>
  <TD>subtotals</TD>
  <TD>131.25</TD>
  <TD>218.00</TD>
  <TD>72.00</TD>
  <TD>421.25</TD>
</TR>
<TR>
  <TD>Totals</TD>
  <TD>196.27</TD>
  <TD>442.00</TD>
  <TD>162.00</TD>
  <TD>800.27</TD>
</TR>
</TABLE>
```

21. Use the *summary* attribute to describe the intent of the design for a non-tabular table.

(a) **Example:**

```
<table summary="Layout table: The first cell contains the navigation bar for the
Web site and the second contains the main content." Border="0" width="100%"
cellspacing="0" cell padding="0">
```

Testing

Test the tables with the JAWS reader to ensure that they read left to right and top to bottom. Use JAWS to read each of the tables. Ensure that all of the information in the table has been properly tagged and can be interpreted correctly.

References

[Identify row and column headers](#)



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

<http://www.w3.org/TR/WCAG10-HTML-TECHS/#tables>

22. Frames

The frameset should be constructed so that the user is not dependent upon visual cues in order to navigate the site. Frames should be titled with text that facilitates frame identification and navigation.

(b) Rationale

Frames need to be titled so that the screen readers can identify them for the visually impaired. Untitled frames make it difficult for the visually impaired to navigate to the content of the Web site.

Techniques

Provide a meaningful *title* attribute for each <FRAME> tag. Give both the frame page and frame source pages meaningful <TITLE> tags. Make sure the <TITLE> tag describes the content of the page.

Example:

```
<HTML>
<HEAD>
<TITLE>Electronic Documents </TITLE>
</HEAD>
<FRAMESET COLS= "10%, 90%" title = "Our library of electronic documents">
  <FRAME src = "logo.html" title = "Company logo frame">
  <FRAME src = "navigation.html" title = "navigation frame">
  <FRAME src = maincontent.html" title = "Company main content">
</NOFRAMES> <A href = "companyindex.html" title = "Company index">
  Select to go an index of the Company's Web site. </A>
</NOFRAMES>
</FRAMESET>
```

Testing

Check that each frame page has a meaningful title in the <TITLE> tag.

Check that every <FRAME> tag has a meaningful *title* attribute.

23. References:

For more information on this topic, please check the following link:

(c) [Frame title](#)

<http://www.w3.org/TR/WCAG10-HTML-TECHS/#frame-names>

24. Scripts, Applets and Plug-ins



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

When pages utilize scripting languages to display content or to create interface elements, the script must provide meaningful text that can be read by assistive technology. If an accurate message cannot be conveyed by the results of the script, provide an equivalent alternative.

Provide a link to a plug-in or applet that complies with the Access Board's standard, §1194.21 (Software Applications and Operating Systems) when a Web page requires an applet, plug-in, or other application to be present on the client system to interpret page content.

(d) Rationale

Assistive technologies may not support some scripts, applets or plug-ins. A screen reader will often read the content of the script as a meaningless jumble of numbers and letters.

Scripts that provide information that are not essential to the understanding or navigation on the Web site can still be used and do not need an equivalent alternative.

Techniques

If the results of a script, applet, or plug-in cannot be accessed by the assistive technology or a meaningful message can not be conveyed by the results of assistive technology, provide an alternate format or method that will be used in addition to, or in place of, the primary Web page, site, application, information, or data. Alternate formats or methods are intended to be equivalent to the primary Web source in both content and functionality and must be updated concurrently with the primary Web source.

When compliance cannot be accomplished in any other way, provide a text-only page equivalent to the primary Web source in both content and functionality and update the text-only page concurrently with the primary Web source.

Scripts

Scripts that dynamically generate content that cannot be conveyed by an assistive technology must have an equivalent alternate format, method or text-only page that can be read by assistive technology. For example, dynamically generated links must be accompanied by redundant text links.

Applets and Plug-ins

Use the *alt* attribute to alternate text descriptions for applets.

Example:

```
<applet code = "gravity.class"  
width = "200" height = "250"alt="Java gravity applet">...</applet>
```



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Determine if the plug-in is accessible to screen readers. If the plug-in is not accessible, provide alternate access to the content. Provide a text description of an object within the <OBJECT>tag.

Example:

```
<OBJECT classid="java:gravity:class"
width = "200" height = "250"> When gravity acts on an object, the weight...</OBJECT>
```

If a plug-in, applet or other application is necessary to access the content of the Web site, provide a hyperlink to the plug-in. The plug-in must comply with the Access Board's standard, §1194.21 (Software Applications and Operating Systems).

Accessible alternate methods might include one-on-one support, telephone support, mailing a printout of the information, fax, audiotape, Braille printout, or download of data in another format.

Java Accessibility

Sun's Java Accessibility Utilities help assistive technologies and interact with applications developed using the Java Accessibility Application Program Interface (API). An application developer can use these APIs directly or indirectly using the Swing toolkit. Developers will be able to create Java Applications capable of interacting with assistive technologies such as screen readers, speech recognition systems, and Braille terminals.

Accessibility-enabled Java applications are not dependent on machines that require assistive technology support. These applications will run on any Java platform with or without assistive technologies. However, to provide access to Java applications, an assistive technology requires more than the Java Accessibility API. It also requires support in locating the components that implement the API; support for being loaded into the Java Virtual Machine; and support for stacking events.

The Java Accessibility Utilities provide this assistance. Note that, developers do not need these utilities to develop accessible applications. The Java Accessibility Utilities are used primarily with assistive technologies.

Testing

Use the JAWS reader to ensure that all content from scripts, applets, or plug-ins is accessible. Ensure that the contents of any alternate content Web pages provide equivalent information and functionality to the original site. Ensure that the applet or plug-in is compliant with the Access Board's standard, §1194.21 (Software applications and Operating Systems). Verify that the content of the alternate format, method, or text-only page is equivalent to the content and functionality of the primary Web source and is updated concurrently with the primary Web source. Ensure that no information has been lost and that the meaning has remained the same.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

References

For more information on this topic, please check the following links:

[Java accessibility](http://www-3.ibm.com/able/accessjava.html)

<http://www-3.ibm.com/able/accessjava.html>

[Sun's JAVA Accessibility Utilities](http://java.sun.com/products/jfc/jaccess-1.2.2/doc/)

<http://java.sun.com/products/jfc/jaccess-1.2.2/doc/>

[Java developer's forum](http://developer.java.sun.com/developer/community/chat/index.html)

<http://developer.java.sun.com/developer/community/chat/index.html>

25. Non-Text Elements

A descriptive text equivalent should be provided for all non-text elements that provide information required for comprehension of the content or that facilitates navigation (images, graphs, charts, or animation).

(e) Rationale

A chart or graph is essentially an image with detailed information. Animation is an image that has movement associated with it. Screen readers cannot interpret images without associated text.

Techniques

Use the *alt* attribute to provide a descriptive text equivalent that summarizes the content of each non-text element.

Example:

```
<img src = http://www.de.state.az.us.com/images/headnav.gif alt= "DES logo">
```

When a short alternative descriptive text is not sufficient to adequately convey the information from a non-text element, use the *longdesc* attribute to provide a longer, more detailed description. This requires the creation of an additional HTML file that stores the more detailed information about the non-text element.

The *longdesc* attribute is placed within an image tag to link the description or data to the image. Although the *longdesc* attribute is not currently accessible by screen readers, future versions will support this attribute. It is suggested that the *longdesc* attribute be utilized now so those Web pages will not require additional remediation. Until this attribute is fully supported, an additional link is required to direct the user to the HTML page that stores the more detailed information about the image.

Example:

```
<img width = "400" height = "263" src = "piechart.gif" longdesc = "chart.htm" alt= "Pie chart displaying percentages of vehicles passing through a toll booth">
```



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

For images that are not used for comprehension, navigation or are redundant, use a blank *alt* attribute to ignore the image. Examples of this include a block of color or a transparent graphic that is used as a filler/spacer, design element, or an image used as a background.

Example:

```
<IMG SRC = http://www.de.state.az.us.com/filler.gif alt="">
```

Another alternative to providing a more detailed description of an image is the "D-Link" (description link).

Example:

```
<a href = "chart.htm" title = "Vehicle Pie Chart"> [D]</a>
```

Provide a text equivalent for animations using the *alt* attribute. If a longer description is necessary, use the *longdesc* attribute or "D-Link" alternative.

If a Microsoft Word document contains an image, JAWS will not read the image. Provide a text description of the image if its description is not contained within the document.

Testing

Mouseover the image to verify that descriptive text appears for each image. Turn images off in the browser. Each image should be replaced with its descriptive text. Click on the links provided by the *longdesc* attribute or "D-Link" to ensure that the user is taken to the appropriate description.

References

For more information on this topic, please check the following link:

[Text equivalents
http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-text-equivalent](http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-text-equivalent)

26. Image Maps

Use client-side image maps, whenever possible, in place of server-side image maps, except when the regions cannot be defined with an available geometric shape. If server-side image maps are used, provide redundant text links for the image map hot spots, except when the regions cannot be defined with an available geometric shape.

(f) Rationale

Screen readers cannot read images. However, image maps are accessible when alternate text is provided for each hot spot area.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

A client-side image map contains all of the information about the image map. It is stored within the HTML document and interpreted through the browser.

The server-side image map is external to the HTML document. A server-side image map requires a script on a Web server that identifies the hot spots and their corresponding uniform resource locators (URL). When a server-side image map is used, browsers cannot indicate to the user, which URL will be followed when a region of the map is activated.

Techniques

Use the *alt* attribute within the and <AREA> tags to provide alternative descriptive text for all areas of an image map.

Example:

```
<IMG src = "Welcome.gif" alt = "areas in the library" usemap = "#map1">
<MAP name = "map1">
<AREA shape = "rect" coords = "0,0,30,30" href = "reference.htm" alt = "Reference">
<AREA shape = "rect" coords = 34,34,100,100 href = "media.html" alt="Media lab">
</MAP>
```

Hot spots, located in a server-side image map, must have redundant text links.

Example:



North America | South America | Africa | Australia

```
<a href = "worldmap.map">
<img src = "worldmap.gif" width180 height = 100 alt = "World Map" border = "0" x-usemap =
"#worldmap" ismap>
</a>
<map name = "worldmap">
<area shape = "rect" coords = "43,39,63,81" href = "http://www.southamerica.com">
<area shape = "rect" coords = "22,9,49,36" href = "http://www.northamerica.com">
<area shape = "rect" coords = "73,26,105,67" href = "http://www.africa.com">
<area shape = "rect" coords = "131,51,156,70" href = "http://www.australia.com">
<area shape = "default" nohref>
</map>
<p>
<a href = "http://www.northamerica.com" title = "North America">North America</a>| <a
href = "http://www.southamerica.com" title = "South America">South America</a>|<a href =
"http://www.africa.com" title = "Africa">Africa </a> | <a href= "http://www.australia.com"
title = "Australia">Australia</a>
```

Testing



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Mouseover the active regions of the client-side image map to ensure that a descriptive text link appears for each hot spot.

Verify that all redundant links work properly.

Reference

For more information on this topic, please check the following links:

[Redundant text links](#)

<http://www.w3.org/TR/WCAG10-TECHS/#tech-redundant-server-links>

27. Multimedia

Provide equivalent alternatives for any multimedia presentation and synchronize them with the presentation.

(g) Rationale

Audio content, without captions or transcripts, is not accessible to the hearing impaired. Videos, with out text descriptions, are not accessible to the visually impaired. In both cases, the information needs to be provided in an alternate format.

Techniques

Provide synchronized captions or transcripts of audio content.

Provide text and audio descriptions of the action occurring in the video.

Provide a descriptive text equivalent for *earcons* or *soundcons* (audio clips), if the information in the audio clip helps the user to interpret the content of that Web site.

Provide a link to any player or plug-in that is required in order to render multimedia. The player or plug-in must comply with the Access Board's standard, §1194.21 (Software Applications and Operating Systems).

Ensure that all alternate formats are updated concurrently with the audio and video Web content.

Testing

Search for all audio and video objects. Ensure that each audio and video object has a corresponding accessible equivalent. Verify that the content of the alternate format is equivalent to the content of the primary audio and video Web content and is updated concurrently with the primary Web source. Ensure that no information has been lost and the meaning has remained the same. Use the JAWS reader to ensure that the player or plug-in is accessible. Ensure that the plug-in or player itself is compliant with the Access Board's standard, §1194.21 (Software Applications and Operating Systems).



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

References

For more information on this topic, please check the following link:

[Synchronize equivalents](http://www.w3.org/TR/WCAG10-TECHS/#tech-synchronize-equivalents)
<http://www.w3.org/TR/WCAG10-TECHS/#tech-synchronize-equivalents>

28. Color

Web pages should be designed so that all information required for navigation or meaning is independent of the ability to identify specific colors.

(h) Rationale

People who cannot differentiate between certain color combinations or have low vision may have difficulty navigating or interpreting content that is dependent upon the ability to identify color.

When foreground and background colors are too close to the same hue, they may not provide sufficient contrast between colors.

Techniques

Identify information in such a way that the message is not conveyed through color alone. For example, do not instruct a user to "Select an item from those listed in green". Instead, relay that information without referring to color.

Use colors and shades that have sufficient contrast.

Testing

Test the document without colors by viewing it with a monochrome monitor. Ensure that the information on the Web site can be viewed when using high contrast appearance settings (i.e., white on black).

References

For more information on this topic, please check the following link:

[Colors](http://www.w3.org/TR/WCAG10-TECHS/#tech-color-convey)
<http://www.w3.org/TR/WCAG10-TECHS/#tech-color-convey>
[Web/Computer color chart for the color blind](http://www.w3.org/TR/WCAG10-TECHS/#tech-color-convey)
<http://www.toledo-bend.com/colorblind/colortable.html>

29. Navigation and Design

Provide a method that will permit users of assistive technology the option to skip repetitive navigation links. Provide descriptive text for all links.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Page design cannot cause the screen or screen element to flicker or blink with a frequency greater than 2 Hertz (Hz) and lower than 55 Hz. When a timed response is required, alert the user and give them sufficient time to indicate that additional time is necessary.

When compliance cannot be achieved, provide an alternate format or method that will be used in addition to, or in place of, the primary Web page, site, application, information, or data. Alternate formats or methods are intended to be equivalent to the primary Web source in both content and functionality and must be updated concurrently with the primary Web source.

When compliance cannot be accomplished in any other way, provide a text-only page equivalent and update it concurrently with the primary Web source.

(i) Rationale

It is common for Web authors to place navigation links at the top, bottom, or side of every new Web page. This technique can render the use of a Web site very difficult for persons using a screen reader because screen readers move through pages reading from top to bottom. The use of repetitive navigation links forces persons with visual impairments to re-read these links when moving to every new page.

A lengthy URL or a hypertext link such as "click here" does not clearly indicate the destination or purpose of the link. Descriptive text links enable assistive technology to provide information that describes the destination or purpose of the link.

Some individuals with photosensitive epilepsy can have a seizure triggered by displays, which flicker or flash at a certain intensity or frequency.

A disability can have a direct impact on the speed with which a person can read and move around a Web page. The Web may expire if a response is not received within a specified time.

Alternate formats, methods, or text-only pages are intended to be equivalent to both the content and functionality, and current with the primary Web source in order to provide the same information and a similar experience.

Techniques

To facilitate easy tracking of page content, insert a link (with text description) before a repetitive navigation link so that users will be able to go directly to the page. This link may either be displayed to all users or be transparent so that it can be read only by a screen reader.

Descriptive text links should state more than *click here*. However, it should not be so wordy that it interferes with efficient browsing.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Example:

Go to the [DES Home Page](#)

Use the *alt* attribute to provide alternative descriptive text for an image that is used as a link.

Example:



Alert the user that a time-out is about to occur. Give the user ample opportunity to answer a prompt asking if the user needs more time. Repeat the process until the user indicates that additional time is not required.

Testing

Examine all repetitive navigation links. Ensure that you insert a link that will allow a user to skip the repetitive navigation links. Check the link to make sure that it takes the user directly to the content of the Web page. Examine links apart from the body of the document. Make sure that descriptive text still makes sense when read out of context by the screen reader. Ensure that a display which flickers or flashes is not within the frequency range of 2 to 55 Hz or flashes per second, has a peak sensitivity at 20 Hz or flashes per second, or has a quick change from dark to light (like strobe lights). Ensure that any time-outs have an option that will allow a user to indicate that additional time is required. Ensure that text-only pages can be read correctly by a screen reader and that the information is equivalent to the primary Web source. Verify that the content of the alternate format, method, or text-only page is equivalent to both the content and functionality of the primary Web source and is updated concurrently with the primary Web source. Ensure that no information has been lost and that the meaning has remained the same.

References

For more information on this topic, please check the following link:

[Avoiding flicker](#)
<http://www.w3.org/TR/WCAG10-TECHS/#tech-avoid-flicker>

30. Graphs and Charts

A chart or graph is essentially an image with detailed information. Some charts or graphs have more information and data in them than others. Text browsers and screen readers cannot convey



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

images to their users, so an equivalent alternative to the essential information that is visually available from the chart or graph must be provided. The amount of alternative information to provide depends on the contextual use of the chart or graph, but generally it should include all the information available to the sighted user.

All charts and graphs should have alternative text. A summary of the content is usually appropriate for the alt attribute of the image. The additional detail data from the chart or graph is not intended to be included in the alt attribute. Instead, use one of the following three techniques: surrounding text, long description, and the D-link. When to use each technique depends on the contextual use of the chart or graph, the importance of the data, visual design considerations, and support in the browsers.

The National Braille Association (NBA) has published a Tape Recording Manual that gives instructions to readers on how to describe complex pictures when recording material for use by blind individuals. The World Wide Web Consortium has made available an [excerpt from the NBA Tape Recording Manual](#) to assist Web authors who are trying to describe complex images on their Web sites. It can also be used to describe complex images in documentation.

(j) Surrounding Description Text

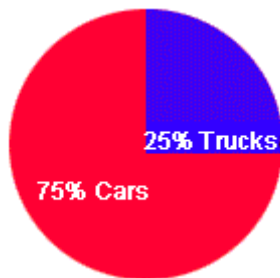
Include a description of the information in the graph or chart as text surrounding the image. For example:

This pie chart shows the percentage of cars and trucks passing through a toll booth in one hour:

75% of the vehicles were cars; 25% of the vehicles were trucks.

```

```



Long Description

This technique is used when it is not desirable to include all the detail about data from the graph in the surrounding text. However, until most browsers support the long description attribute, it is necessary to use the D-link (description link) technique with the longdesc attribute. IBM Home Page Reader currently supports longdesc. The CODA Technologies company provides tools such as [PopChart™ that makes accessible charts and graphs](#) with descriptive text.

Use the longdesc attribute to point to a URL that contains a detailed description and data. For example, because the information about the colors of the pie chart is only available from the visual image, a separate description is provided that includes the information about the colors of the pie chart. A blind math student, or a student using a text-only browser, would be able to answer the question using the separate description.



Version 1.0

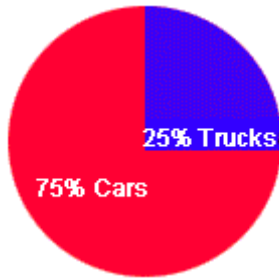
Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Math student: From the chart determine which vehicle type is represented by the smaller blue wedge:

```

```



The longdesc points to the separate description file that includes the following sample HTML:

```
<p>Pie Chart Description</p>
```

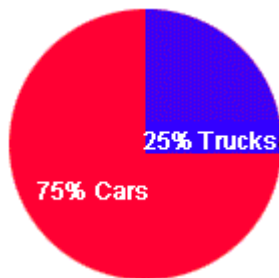
```
<p>This pie chart shows the percentage of cars and trucks out of the total vehicles passing through a tollbooth in one hour. Cars are represented by the larger red wedge and trucks are represented by the smaller blue wedge. 75% of the vehicles were cars; 25% of the vehicles were trucks. The total number of cars and trucks is not displayed on the pie chart.</p>
```

D-Link

This technique is used when it is not desirable to include all the detail about the graph or chart in the surrounding text. Use the description link or D-link to access a URL containing the detailed description of the graph or chart. The current convention to minimize the affect on the visual appearance of the page is to use the letter D as the link text.

In the following example, information about the size and shape of the wedges that make up the pie chart are only available from the visual image. A detailed description needs to be included so a blind math student (or a student using a text-only browser) will understand the visual concept and be better able to answer the question.

```
  
<a href="piechart_desc.html">D</a>
```



This example uses an invisible image which does not change the visible appearance of the page at all yet will be read by a screen reader.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

```

<a href="piechart_desc.html"></a>
```

31. Style Sheets

Documents must be organized so that they are readable without requiring an associated style sheet. Documents must be constructed so that the user is not dependent upon style sheets to interpret the content of the Web page. This does not prohibit the use of style sheets.

(k) Rationale

If not organized properly, style sheets may make it difficult for Web pages to be read accurately by browsers that do not support style sheets or in a browser where a user has disabled support for style sheets.

Techniques

Arrange style commands so that the contents make sense and read in the correct order without the associated style sheets.

Make sure that the Web page is useable when the style sheets have been disabled in the browser or the user has elected to activate their user-developed style sheets. If the content has changed or is not useable, provide an alternative Web page.

The following two examples show the use of style commands. The first example is incorrect because it only uses horizontal positioning. In the second example, both horizontal and vertical positioning are used which, leads to proper ordering of the sentence, regardless of whether the style sheets are disabled.

Example:

```
<STYLE TYPE="text/css">
.part1 /* The quick */ {padding-left 0;
color: red; font-size: 14pt;
font-family: copperplate gothic bold, fantasy, sans-serif }
.part2 /* brown fox */ { padding-left: 100px; color: brown; font-size: 10pt;
font-family: times new roman, desdemona, serif }
.part3 /* jumped over */ { padding- left: 350px;
color: purple; font size: 18pt;
font-family: desdemona, times new roman, serif }
.part4 /* the lazy dog */ { padding-left: 350px;
color: blue; font size: 24pt;
font-family: fantasy, copperplate gothic bold, sans serif }
</STYLE>
<DIV class=part4>the lazy dog.</DIV>
```



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

```
<DIV class=part1>The quick</DIV>
<DIV class=part3>jumped over</DIV>
<DIV class=part2>the brown fox</DIV>
```

In the previous example, if a user elected to use their own style sheets or if style sheets were disabled in the browser, the sentence would read incorrectly:

Example:

```
<STYLE TYPE ="text/css">
<!--
.part1 /* The quick */ {color: red; font-size: 14 pt;
padding-left 0; margin-top 40px;
font-family: copperplate gothic bold, fantasy, sans-serif }
.part2 /* brown fox */ { color: brown;font-size:10 pt;
padding-left: 100px; margin-top:30px;
font-family: times new roman, desdemona, serif }
.part3 /* jumped over */ { color: purple; font-size: 18pt;
padding- left: 200px; margin-top: -60px;
font-family: desdemona, times new roman, serif }
.part4 /* the lazy dog */ { color: blue; font size: 24 pt;
padding-left: 350px;
margin-top: -100px; margin-bottom: 100px;
font-family: fantasy, copperplate gothic bold, sans serif }
>
</STYLE>

<DIV class=part1>The quick</DIV>
<DIV class=part2>the brown fox</DIV>
<DIV class=part3>jumped over</DIV>
<DIV class=part4>the lazy dog. </DIV>
```

In the above example, if a user elected to used their own style sheets or if style sheets were disabled in the browser, the sentence would read correctly:

The quick
brown fox
jumped over
the lazy dog.

Testing

Turn the style sheets off in the browser. Ensure that all content can be understood. Create a blank style sheet and configure your browser to use this style sheet to override all non-user defined style sheets.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

References

For more information on this topic, please check the following link:

[Style Sheet organization](#)

<http://www.w3.org/TR/WCAG10-TECHS/#tech-order-style-sheets>

32. Flash Movies & Animation

Macromedia Flash provides a powerful set of tools for creating animations and interactive content and applications. These tools allow designers and developers to create demonstrations of complex visual relationships, engaging interactions in buttons and movies, and attractive interfaces in websites and web applications. Most assistive technologies, however, are not designed with these types of animations in mind. While working in Macromedia Flash, designers and developers need to be mindful of the ways in which assistive technologies such as screen readers will interact with their movies. This brief section covers the following information on how to optimize accessibility when creating animation:

Understanding how assistive technologies handle animation

Providing text equivalents for entire movies

Making child objects inaccessible, Avoid constant motion

(I) Understanding How Assistive Technologies Handle Animation

When a screen reader encounters a piece of Macromedia Flash content, the screen reader loads the current state of the movie and notifies the user. With the Window-Eyes screen reader, the user hears, "Loading...load done." Once a piece of content has been read, the screen reader moves on to read other parts of the Macromedia Flash content and the rest of the page.

A unique feature of Macromedia Flash content is that it may change over time. As the content changes, Macromedia Flash Player 7 sends a signal to the screen reader notifying it that there has been a change. When the screen reader receives this notification, it automatically returns to the top of the page and begins reading it again.

The following example illustrates the serious implications of Macromedia Flash content created without consideration for users of screen readers. A poorly designed banner ad placed at the top of the page might loop constantly through a few frames. When Macromedia Flash Player encounters this banner, it will send repeated notifications to the screen reader of changes in the content, and the screen reader will continually return to the top of the page. This problem can seriously erode the experience for screen reader users.

To address this specific issue, Macromedia worked with GW Micro to create a 'Halt Flash events' keystroke (Alt-Shift-M) for the Window-Eyes screen reader. This keystroke allows a screen reader user to suspend Macromedia Flash notifications on the page. Pressing the keystroke again allows the user to resume Macromedia Flash notifications.



Version 1.0

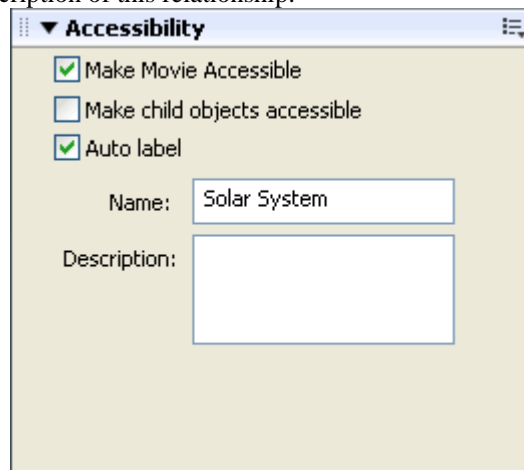
Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

33. Providing Text Equivalents for Entire Movies

A better solution for the handling of animation is simply to hide the animation from assistive technologies. Macromedia Flash MX 2004 allows designers and developers to assign a text equivalent for an entire movie or for a collection of objects within a movie. Designers and developers might choose to provide a text equivalent for Macromedia Flash content for one of two reasons.

Animations are often used to illustrate visual relationships among elements on the screen. Adding text equivalents to the individual elements may not provide a sufficient description of the relationships among the elements. For example, in an animation of the solar system, a designer/developer might add text equivalents to the planets; however, these text equivalents would not convey information about how the planets move in relation to one another. A text equivalent for the entire movie could provide a better description of this relationship.



34. Making Child Objects Inaccessible

Notice in the example above that the option “Make child objects accessible” has been deselected. This serves two valuable purposes. First, it serves to group these elements formally as a single element. From a testing standpoint, this makes the movie easier to evaluate for accessibility. Second and more importantly, it hides the motion from screen readers. By making the child objects of this sample movie inaccessible, the designer/developer is effectively telling the screen reader to ignore everything within the movie. As a result, this movie will not send the constant updates to the screen reader that cause the screen reader to keep looping.

35. Avoid Constant Motion

An important consideration for accessible animation relates to constant motion on the screen. While it is quite popular to include motion in movies as part of transitions and loading sequences, it is important that these animations settle to a static screen once the page loads. For people with learning disabilities, motion on the screen can be distracting and might even make other elements unreadable.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

36. Accessibility and Macromedia Flash Player

With added support for scripting accessibility information, support for accessible components, enhanced integration with assistive technologies, and increased keyboard access, Macromedia Flash Player 7 delivers a powerful new array of features. Together with Macromedia Flash MX 2004 and Macromedia Flash Professional 2004, Macromedia Flash Player 7 offers developers the most robust set of tools available for developing accessible rich media. In addition, Macromedia Flash MX 2004 introduces new features that allow third-party developers to create a variety of extensions that make authoring and delivering accessible rich media even easier.

37. Enhanced Accessibility

With integrated support for Microsoft Active Accessibility (MSAA), Macromedia Flash Player 7 makes content available via screen access technologies such as Window-Eyes from GW Micro and JAWS from Freedom Scientific. Macromedia Flash Player 7 includes support for complex, dynamic sites that update their content via ActionScript. Macromedia Flash Player 7 also addresses a problem common among ActiveX controls that prevents users with mobility impairments from moving easily between Macromedia Flash and HTML content.

To help designers and developers ensure that Rich Internet Applications are accessible, Macromedia Flash Player 7 and Macromedia Flash MX 2004 include support for updating and generating the accessibility properties via ActionScript. This enables applications to update accessibility information as the content changes.

Additionally, the Accessibility Resource Center provides a number of new documents offering tips and tricks to designers and developers seeking to create accessible Macromedia Flash content. These documents include information on adding text equivalents to objects in Macromedia Flash MX 2004, marking up forms, working with text, and adding animation. This site will be frequently updated with new tools and information to support accessible design in Macromedia Flash MX 2004.

System Requirements

To access Macromedia Flash content using a screen reader, users will need to have the Macromedia Flash Player 6 or better installed on their machines. To download the latest version of Macromedia Flash Player, click the link below:

<http://www.macromedia.com/go/getflashplayer/>

Users will also need a screen reader that has integrated the Macromedia Flash Player implementation of MSAA. As Macromedia Flash Player 7 is released, these include **Window-Eyes** from GW Micro and **JAWS** from Freedom Scientific. More information about the Window-Eyes screen reader and a demo version are available at www.gwmicro.com.



Version 1.0

Department of Economic Security

Subject: Web Accessibility Guidelines – For Developers

Finally, screen reader users will need to access Macromedia Flash content using the Microsoft Internet Explorer browser. This is the only browser with support for MSAA.

Accessing Macromedia Flash Content

When a screen reader encounters Macromedia Flash content, it typically announces the beginning and end of each Macromedia Flash movie. Otherwise, the content functions the same way as HTML content on a web page. As an example, the simple Macromedia Flash movie below contains seven elements. There is a title at the top that reads, "Electronic Registration Card." There are two input fields, labeled "Name" and "Address," respectively. Finally, there is a button at the bottom with a label that reads, "Send."

The screenshot shows a window titled 'form.swf' with a light blue background. At the top, the text 'electronic registration' is displayed in a light blue, italicized font. Below this, there are two input fields. The first is labeled 'Name:' and the second is labeled 'Address:'. Both labels are in a bold, black font. To the right of each label is a rectangular input field. At the bottom right of the form, there is a 'send' button with a blue sphere icon next to it.

A screen reader would likely read this movie as follows: "Electronic registration card. Textfield name. Textfield address. Button send." The word "textfield" is a cue to the user that there is a field that needs to be completed. The word "button" is a cue that there is a button on the page. A screen reader user would complete these fields and activate the button in the same way he or she would in an HTML document. By default, Macromedia Flash Player reads text elements, buttons, and input text in Macromedia Flash content. Using Macromedia Flash MX 2004, designers and developers can assign text equivalents for movie clips and even entire movies. With the new accessible components, designers and developers can easily create accessible web applications. Even the most complex movies can be made accessible quickly and easily.

38. Appendix

(m) Validation and Testing

Automated validation methods are generally rapid and convenient, but cannot identify all accessibility issues. Human review can help ensure clarity of language and ease of navigation.

Begin using validation methods at the earliest stages of development. The earlier you identify accessibility issues, the easier it is to correct and avoid these issues.

The following are some important validation methods:

Use the JAWS reader to determine whether all information can be read and interpreted correctly.

For best results, use JAWS in conjunction with Internet Explorer 5.x.

Validate the HTML, using a tool such as the W3C's [HTML Validation Service](http://validator.w3.org/) (<http://validator.w3.org/>).

Validate style sheets, using a tool such as the W3C's [CSS Validation Service](http://jigsaw.w3.org/css-validator/) (<http://jigsaw.w3.org/css-validator/>).

Use the browser without a mouse.

Use the browser with the following options turned on and off:

Sounds.

Graphics.

Style sheets.

39. W3C's Web Accessibility Tutorial

Check this link for the [W3C's Web Accessibility Tutorial](http://www.w3.org/WAI/wcag-curric/) (<http://www.w3.org/WAI/wcag-curric/>).

40. Web Accessibility Checklist

Use this as a tool for high-level guidance in determining if a Web site is compliant or accessible.

Web Accessibility Checklist			
Guidance		YES NO NA	Comments
1	PDF Files: An alternate format should be provided either in addition to an Adobe Portable Format (PDF) file or in place of the PDF file.		
2	Forms: When electronic forms are designed to be completed online, the form must allow access via assistive technology to information, field elements, and functionality (e.g., associated controls required for completion and submission of the form, including directions and cues).		
3	Tables: Tables must be constructed so that they can be easily interpreted by all users and communicate the intent of the author. Identify row and column headers for data tables. Associate data cells with their headers for all tables that have two or more logical levels of row or column headers. Non-tabular tables may be used for layout if the table uses appropriate HTML to summarize the intent of the design. Make sure that tables make sense when read from left to right, and from top to bottom.		
4	Frames: The frameset should be constructed so that the user is not dependent upon visual cues to navigate the site. Frames should be titled with text that facilitates frame identification and navigation.		
5	Scripts, Applets, and Plug-ins: When pages utilize scripting languages to display content or to create interface elements, the script must provide meaningful text that can be read by assistive technology. If an accurate message cannot be conveyed by the results of the script, provide an equivalent alternative. Provide a link to a plug-in or applet that complies with the Access Board's standard, §1194.21 (Software Applications and Operating Systems) when a Web page requires an applet, plug-in or other application to be present on the client system to interpret page content.		
6	Non-Text Elements: A descriptive text equivalent should be provided for all non-text elements (e.g., images, graphs, charts, or animation) that provide information required for comprehension of the content or that facilitates navigation.		
7	Image Maps: Use client-side image maps, whenever possible, in place of server-side image maps, except when the regions cannot be defined with an available geometric shape. If server-		

Web Accessibility Checklist			
Guidance		YES NO NA	Comments
	side image maps are used, provide redundant text links for the image map hot spots, except when the regions cannot be defined with an available geometric shape.		
8	Multimedia: Provide equivalent alternatives for any multimedia presentation and synchronize them with the presentation.		
9	Color: Web pages should be designed so that all information required for navigation or meaning is independent of the ability to identify specific colors.		
10	Navigation and Design: Provide a method that will permit users of assistive technology the option to skip repetitive navigation links. Provide descriptive text for all links. Page design cannot cause the screen or screen element to flicker or blink with a frequency greater than 2 Hz and lower than 55Hz. When a timed response is required, alert the user and give them the sufficient time to indicate that additional time is necessary. When compliance cannot be achieved, provide an alternate format or method that will be used in addition to or in place of the primary Web page, site, application, information or data. Alternate formats or methods are intended to be equivalent to the primary Web source both in content and functionality and must be updated concurrently with the primary Web source. When compliance cannot be accomplished in any other way, provide a text-only page equivalent and update it concurrently with the primary Web source.		
11	Style Sheets: Documents must be organized so that they are readable with out requiring an associated style sheet. Documents must be constructed so that the user is not dependent upon style sheets in order to interpret the content of the Web page. This does not prohibit the use of style sheets.		

41. CONCLUSION

Following the above guidelines and web page coding standards we can make our website WIA compliant. And thus JAWS v3.7 and above will not have any problems what so ever in reading a web application having FRAMES or FRAMES less solution.